VIM Desktop Reference

Basic movement

character left, right; line up, down	`h l k j`
word/token left, right	`b w`
end of word/token left, right	`ge e`
beginning of previous, next section	`{ }`
beginning of previous, next sentence	`( )`
beginning, middle of line	`0 gm`
first, last character of line	`^ $`
line n, default the last, first	n`G` n`gg`
percentage n of the file (n must be provided)	n`%`
column n of current line	n`|`
match of next brace, bracket, comment,`#define`	`%`
line n from start, bottom of window	n`H` n`L`
middle line of window	`M`

Insertion & replace → insert mode

insert before, after cursor	`i a`
insert at beginning, end of line	`I A`
insert text in first column	`gI`
open a new line below, above the current line	`o O`
replace character under cursor with c	`r`c
like r, but without affecting layout	`gr`c
replace characters starting at the cursor	`R`
like R, but without affecting layout	`gR`
change text of movement command m	`c`m
change current line	`cc or S`
change to the end of line	`C`
change one character and insert	`s`
switch case and advance cursor	`~`
switch case of movement command m	`g~`m
lowercase, uppercase text of movement m	`gu`m `gU`m
shift left, right text of movement m	`<`m `>`m
shift n lines left, right	n`<` n`>`

Deletion

delete character under, before cursor	`x X`
delete text of movement command m	`d`m
delete current line, to the end of line	`dd D`
join current line with next, without space	`J gJ`
delete range r lines	`:`r`d`
delete range r lines into register x	`:`r`d`x

Insert mode

insert char c literally, decimal value n	`^V`c `^V`n
insert previously inserted text	`^A`
same as`^A`and stop insert command mode	`^@`
insert content of register x, literally	`^R`x `^R^R`x
text completion before, after cursor	`^N ^P`
delete word before cursor	`^W`
delete all inserted character in current line	`^U`
shift left, right one shift width	`^D ^T`
enter digraph $\{c_1\ ,\ c_2\}$	`^K`c_1 c_2 or c_1c_2
execute c in temporary command mode	`^O`c
scroll up, down	`^X^E ^X^Y`
abandon edition command mode	`esc or ^[`

Copying

use register x for next delete, yank, put	`"`x
show the content of all registers	`:reg`
show the content of registers x	`:reg` x
yank the text of movement command m	`y`m
yank current line into register	`yy or Y`
put register after, before cursor position	`p P`
like p, P with indent adjusted	`]p [p`
like p, P leaving cursor after new text	`gp gP`

Advanced insertion

perform rot13 encoding on movement m	`g?`m
$+n$, $-n$ to number under cursor	n`^A` n`^X`
format lines of movement m to fixed width	`gq`m
center lines in range r to width w	`:`r`ce` w
left align lines in range r with indent i	`:`r`le` i
right align lines in range r to width w	`:`r`ri` w
filter lines of movement m through command c	`!`mc
filter n lines through command c	n`!!`c
filter range r lines through command c	`:`r`!`c

Visual mode

start/stop highlighting characters, lines, block	`v V ^V`
exchange cursor position with start of highlighting	`o`
start highlighting on previous visual area	`gv`
select a word, a sentence, a section	`aw as ap`
select a block ( ), a block	`ab aB`

Undoing, repeating & registers

undo last command, restore last changed line	`u U`
repeat last changes, redo last undo	`^R`
repeat last changes with count replaced by n	n`.`
record, append typed characters in register c	`q`c `q`C
stop recording	`q`
execute the content of register c	`@`c
repeat previous @ command	`@@`
execute register c as an Ex command	`:@`c
execute Ex command c on range r where pattern p matches	`:`r`g/`p`/`c

Complex movement

line up, down on first non-blank character	`- +`
space-separated word left, right	`B W`
end of space-separated word left, right	`gE E`
down $n-1$ line on first non-blank character	n`_`
beginning of screen line	`g0`
first, last character of screen line	`g^ g$`
screen line up, down	`gk gj`
next, previous occurence of character c	`f`c `F`c
before next, previous occurence of c	`t`c `T`c
repeat last `fFtT`, in opposite direction	`; ,`
start of section* backward, forward	`[[ ]]`
end of section* backward, forward	`[] ][`
unclosed (,) backward, forward	`[( ])`
unclosed , backward, forward	`[{ ]}`
start of backward, forward Java method	`[m ]m`
unclosed `#`directives ,backward, forward	`[# ]#`
start, end of `/* */` backward, forward	`[* ]*`

Search & substitution

search forward, backward for s	`/`s `?`s
search fwd, bwd for s with offset o	`/`s`/`o `?`s`?`o
repeat forward last search	`n or /`
repeat backward last search	`N or ?`
search backward, forward for word under cursor	`# *`
same, but also find partial matches	`g# g*`
local, global definition of symbol under cursor	`gd gD`
substitute f by t in range r	`:`r`s/`f`/`t`/`x
x : `g`–all occurrences, `c`–confirm changes	
repeat substitution with new r & x	`:`r`s` x

Special characters in search patterns

any single character, start, end of line	. ^ $
start, end of word	\< \>
a single character in range $c_1 \cdots c_2$	[$c_1 - c_2$]
a single character not in range	[^$c_1 - c_2$]
an identifier, keyword; excl. digits	\i \k \I \K
a file name, printable char.; excl. digits	\f \p \F \P
a white space, a non-white space	\s \S
esc , tab , return, backspace	\e \t \r \b
match 0..1, 0.., 1.. of preceding atoms	\= * \+
separate two branches	\\|
group patterns into an atom	(or) \(\)
the whole matched pattern, *n*th () group	\& \n
next character made upper, lowercase	\u \l

Offsets in search commands

n line downward in column 1	*n* or +*n*
n line upward in column 1	−*n*
n characters right, left to end of match	e+*n* e-*n*
n characters right, left to start of match	s+*n* s-*n*
execute search command *sc* next	;*sc*

Marks and motions

mark current position with mark *c*	m*c*
go to mark *c* in current, *C* in any file	‘*c* ‘*C*
go to last exit position	‘0 · · · 9
go to position before jump, at last edit	‘‘ ‘"
go to start, end of previously operated text	‘[‘]
print the active marks list	:marks
print the jump list	:jumps
go to *n*th older position in jump list	*n*^O
go to *n*th newer position in jump list	*n*^I

Key mapping & abbreviations

map *c* ↦ *e* in normal & visual mode	:map *c* *e*
map *c* ↦ *e* in insert & cmd-line mode	:map! *c* *e*
remove mapping *c*	:unmap *c* :unmap! *c*
write current mappings, settings to file *f*	:mk *f*
add abbreviation for *c* ↦ *e*	:ab *c* *e*
show abbreviations starting with *c*	:ab *c*
remove abbreviation *c*	:una *c*

Tags

jump to tag *t*	:ta*t*
jump to *n*th newer tag in list	:*n*ta
jump to the tag under cursor, return from tag	^] ^T
list matching tags and select one for jump	:ts *t*
jump to tag or select one if multiple matches	:tj *t*
print tag list	:tags
jump back from, to *n*th older tag	:*n*po :*n*^T
jump to last matching tag	:tl
preview tag under cursor, tag *t*	^W} :pt *t*
split window and show tag under cursor	^W]
close tag preview window	^Wz or :pc

Scrolling & multi-windowing

scroll line up, down	^E ^Y
scroll half a page up, down	^D ^U
scroll page up, down	^F ^B
set current line at top of window	zt or z
set current line at center of window	zz or z
set current line at bottom of window	zb or z-
scroll one character to the right, left	zh zl
scroll half a screen to the right, left	zH zL
split window in two	^Ws or :split
create new empty window	^Wn or :new
make current window one on screen	^Wo or :on
move to window below, above	^Wj ^Wk
move to window below, above (wrap)	^Ww ^W^W

Ex commands

edit file *f* , unless changes have been made	:e *f*
edit file *f* always (by default reload current)	:e! *f*
write file and edit next, previous one	:wn :wN
edit next, previous file in list	:n :N
write range *r* to current file	:*r*w
write range *r* to file *f*	:*r*w *f*
append range *r* to file *f*	:*r*w>>*f*
quit and confirm, quit and discard changes	:q :q!
write to current file and exit	:wq or :x or ZZ
recall commands starting with current	up down
insert content of file *f* below cursor	:r *f*
insert output of command *c* below cursor	:*r*! *c*
open a window for each file in the argument list	:all
display the argument list	:args

Ex ranges

separates two lines numbers, set to first line	, ;
an absolute line number *n*	*n*
the current line, the last line in file	. $
entire file, visual area	% *
position of mark *t*	’*t*
the next, previous line where *p* matches	/*p*/ ?*p*?
+*n*,−*n* to the preceding line number	+*n* −*n*

Folding

create fold of movement *m*	zf*m*
create fold for range *r*	:*r*fo
delete fold at cursor, all in window	zd zE
open, close one fold; recursively	zo zc zO zC
move to start, end of current open fold	[z]z
move down, up to start, end of next fold	zj zk

Miscellaneous

start shell, execute command *c* in shell	:sh :!*c*
lookup keyword under cursor with man	K
start make, read errors and jump to first	:make
display the next, previous error	:cn :cp
list all errors, read errors from file	:cl :cf
redraw screen, show filename and position	^L ^G
show cursor column, line, and character position	g^G
show ASCII value of character under cursor	ga
open file which filename is under cursor	gf
redirect output to file *f*	:redir>*f*
save view configuration [to file *f*]	:mkview [*f*]
load view configuration [from file *f*]	:loadview [*f*]
unmapped keys	^@ ^K ^ \ Fn ^Fn
